IN THE CLAIMS

Claims 1-36 are pending, as indicated below:

1. (Original) A method for merging one or more hierarchical trees at runtime comprising:

examining one or more nodes in each of said hierarchical trees;

determining if there are one or more sets of equivalent nodes in said hierarchical trees;

picking one or more winning nodes from each of said sets of equivalent nodes; and

storing one or more reference nodes to said winning nodes.

2. (Original) The method of claim 1 wherein said hierarchical trees are document object model (DOM) trees.

3. (Original) The method of claim 1 further comprising:

printing a merged tree using said reference nodes.

4. (Original) The method of claim 1 wherein said reference nodes are one or more pointers.

5. (Original) The method of claim 1 wherein said reference nodes are one or more Java references.

6. (Original) The method of claim 1 wherein said picking further comprises:

examining one or more priorities associated with one or more members in each set of said equivalent nodes; and

selecting said winning node as said member with a highest of said priorities.

7. (Original) The method of claim 1 further comprising

generating one or more shallow clones for said winning nodes; and

adding said shallow clones to a merged tree.

8. (Original) The method of claim 1 wherein said hierarchical trees include a group tree, a user tree, and an admin tree.

9. (Original) The method of claim 2 wherein said DOM trees are eXtensible Markup Language (XML) DOM trees.

10. (Original) A system for merging one or more hierarchical trees at runtime comprising:

one or more nodes in each of said hierarchical trees configured to be examined;

one or more sets of equivalent nodes in said hierarchical trees configured to be located if said sets of equivalent nodes exist;

one or more winning nodes configured to be picked from each set of said equivalent nodes; and

one or more reference nodes to said winning nodes configured to be stored.

11. (Original) The system of claim 10 wherein said hierarchical trees are document object model (DOM) trees.

12. (Original) The system of claim 10 further comprising:

a merged tree configured to be printed using said reference nodes.

13. (Original) The system of claim 10 wherein said reference nodes are one or more pointers.

14. (Original) The system of claim 10 wherein said reference nodes are one or more Java references.

15. (Original) The system of claim 10 wherein further comprising:

one or more priorities associated with one or members in each set of said equivalent nodes configured to be examined wherein said winning node is selected as said member with a highest of said priorities.

16. (Original) The system of claim 10 further comprising

one or more shallow clones for said winning nodes configured to be generated wherein said shallow clones are added to a merged tree.

17. (Original) The system of claim 10 wherein said hierarchical trees include a group tree, a user tree, and an admin tree.

18. (Original) The system of claim 11 wherein said DOM trees are eXtensible Markup Language (XML) DOM trees.

19. (Original) A computer program product comprising:

a computer usable medium having computer readable program code embodied therein configured to merge one or more hierarchical trees at runtime comprising:

computer readable code configured to cause a computer to examine one or more nodes in each of said hierarchical trees;

computer readable code configured to cause a computer to determine if there are one or more sets of equivalent nodes in said hierarchical trees;

computer readable code configured to cause a computer to pick one or more winning nodes from each set of said equivalent nodes; and

computer readable code configured to cause a computer to store one or more reference nodes to said winning nodes.

20. (Original) The computer program product of claim 19 wherein said hierarchical trees are document object model (DOM) trees.

21. (Original) The computer program product of claim 19 further comprising: computer readable code configured to cause a computer to print a merged tree using said reference nodes.

22. (Original) The computer program product of claim 19 wherein said reference nodes are one or more pointers.

23. (Original) The computer program product of claim 19 wherein said reference nodes are one or more Java references.

24. (Original) The computer program product of claim 19 wherein said computer readable code configured to cause a computer to pick further comprises:

computer readable code configured to cause a computer to examine one or more priorities associated with one or more members in each set of said equivalent nodes; and

computer readable code configured to cause a computer to select said winning node as said member with a highest of said priorities.

25. (Original) The computer program product of claim 19 further comprising computer readable code configured to cause a computer to generate one or more shallow clones for said winning nodes; and

computer readable code configured to cause a computer to add said shallow clones to a merged tree.

26. (Original) The computer program product of claim 19 wherein said hierarchical trees include a group tree, a user tree, and an admin tree.

27. The computer program product of claim 20 wherein said DOM trees are eXtensible Markup Language (XML) DOM trees.

28. (Original) An apparatus for merging one or more hierarchical trees at runtime comprising:

means for examining one or more nodes in each of said hierarchical trees;

means for locating one or more sets of equivalent nodes in said hierarchical trees, if said sets of equivalent nodes exist;

means for picking one or more winning nodes from each set of said equivalent nodes; and

means for storing one or more reference nodes to said winning nodes.

29. (Original) The apparatus of claim 28 wherein said hierarchical trees are document object model (DOM) trees.

30. (Original) The apparatus of claim 28 further comprising:

means for printing a merged tree using said reference nodes.

31. (Original) The apparatus of claim 28 wherein said reference nodes are one or more pointers.

32. (Original) The apparatus of claim 28 wherein said reference nodes are one or more Java references.

33. (Original) The apparatus of claim 28 further comprising:

means for examining one or more priorities associated with one or members in each set of said equivalent nodes; and

means for selecting said winning node as said member with a highest of said priorities.

34. (Original) The apparatus of claim 28 further comprising

means for generating one or more shallow clones for said winning nodes wherein said shallow clones are added to a merged tree.

35. (Original) The apparatus of claim 28 wherein said hierarchical trees include a group tree, a user tree, and an admin tree.

36. (Original) The apparatus of claim 29 wherein said DOM trees are eXtensible Markup Language (XML) DOM trees.